Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Original) In a data processing network including distributed processing units, a method comprising:

obtaining a respective utilization value of each distributed processing unit;

applying a mapping function to the respective utilization value of said each distributed processing unit to obtain a respective weight for said each distributed processing unit; and

using the respective weights for the distributed processing units for distributing work requests to the distributed processing units so that the respective weight for said each distributed processing unit specifies a respective frequency at which the work requests are distributed to said each distributed processing unit.

2. (Original) The method as claimed in claim 1, wherein the respective utilization value of said each distributed processing unit is a percentage of saturation of said each distributed processing unit.

3. (Original) The method as claimed in claim 1, wherein said each distributed processing unit collects statistics for calculation of the respective utilization value of said each distributed processing unit.

4. (Original) The method as claimed in claim 1, wherein statistics for calculation of the respective utilization value of said each distributed processing unit are collected from said each distributed processing unit.

5. (Original) The method as claimed in claim 1, wherein the respective weight for said each distributed processing unit is programmed into a mapping table, and the mapping function is applied to the respective utilization value of said each distributed processing unit to obtain the respective weight for said each distributed processing unit by indexing the mapping table with the respective utilization value of said each distributed processing unit to obtain the respective weight for said each distributed processing unit.

6. (Original) The method as claimed in claim 1, wherein the mapping function is selected to provide weights estimated to cause a balancing of loading upon the distributed processing units.

7. (Original) The method as claimed in claim 1, wherein the respective weights are used for weighted round-robin load balancing of the work requests upon the distributed processing units.

8. (Original) The method as claimed in claim 7, wherein the weighted round-robin load balancing performs round-robin load balancing when the weights are equal.

9. (Original) The method as claimed in claim 1, wherein the respective weights for the distributed processing units are used for distributing work requests to the distributed processing units by creating a distribution list containing entries indicating the distributed processing units, the respective weight for said each distributed processing unit specifying the number of the entries indicating said each distributed processing unit, and by randomizing the distribution list, and accessing the randomized distribution list for distributing the work requests to the distributed processing units as indicated by the entries in the randomized distribution list.

10. (Currently amended) The method as claimed in claim [[1]] 9, which includes re-randomizing the distribution list for re-use once the end of the distribution list is reached during the distribution of the work requests to the distributed processing units as indicated by the entries in the randomized distribution list.

11. (Original) In a data processing network including distributed processing units, a method comprising:

obtaining a respective utilization value of each distributed processing unit;

applying a mapping function to the respective utilization value of said each distributed processing unit to obtain a respective weight for said each distributed processing unit;

using the respective weights for the distributed processing units for producing a distribution list for distributing work requests to the distributed processing units for load balancing of the work requests upon the processing units, and repetitively randomizing the distribution list during the distribution of the work requests to the distributed processing units.

12. (Original) In a data processing network including a network file server and a plurality of virus checking servers, a method comprising:

the network file server obtaining a respective utilization value of each virus checking server, the respective utilization value of said each virus checking server indicating a percentage of saturation of said each virus checking server;

the network file server applying a mapping function to the respective utilization value of said each virus checking server to obtain a respective weight for said each virus checking server; and

the network file server using the respective weights for the virus checking servers for weighted round-robin load balancing of virus checking requests from the network file server to the virus checking servers.

13. (Original) The method as claimed in claim 12, wherein said each virus checking server collects statistics for calculation of the respective utilization value of said each virus checking server.

14. (Original) The method as claimed in claim 12, wherein the respective weight for said each virus checking server is programmed into a mapping table, and the network file server

indexes the mapping table with said each respective utilization value to obtain the respective weight for said each virus checking server.

15. (Original) The method as claimed in claim 12, wherein the weighted round-robin load balancing performs round-robin load balancing when the weights are equal.

16. (Original) The method as claimed in claim 12, wherein the respective weights for the virus checking servers are used for weighted round-robin load balancing of virus checking requests from the network file server to the virus checking servers by creating a distribution list containing entries indicating the virus checking servers, the respective weight for said each virus checking server specifying the number of the entries indicating said each virus checking server, and by randomizing the distribution list, and accessing the randomized distribution list for distributing the virus checking requests from the network file server to the virus checking servers as indicated by the entries in the randomized distribution list.

17. (Original) The method as claimed in claim 16, which includes re-randomizing the distribution list for re-use once the end of the distribution list is reached during the distributing of the work requests to the virus checking servers as indicated by the entries in the randomized distribution list.

18. (Original) The method as claimed in claim 16, wherein the network file server obtains the utilization values of the virus checking servers at the start of a heartbeat interval,

randomizes the distribution list repetitively during use of the distribution list for load balancing of virus checking requests during the heartbeat interval, obtains new utilization values of the virus checking servers at the start of a following heartbeat interval, and produces a new distribution list from the new utilization values of the virus checking servers for load balancing of virus checking requests during the following heartbeat interval.

19. (Original) A data processing system comprising distributed processing units and a processor coupled to the distributed processing units for distributing work requests to the distributed processing units, the processor being programmed for:

obtaining a respective utilization value of each distributed processing unit;

applying a mapping function to the respective utilization value of said each distributed processing unit to obtain a respective weight for said each distributed processing unit; and

using the respective weights for the distributed processing units for distributing work requests to the distributed processing units so that the respective weight for said each distributed processing unit specifies a respective frequency at which the work requests are distributed to said each distributed processing unit.

20. (Original) The data processing system as claimed in claim 19, wherein the respective utilization value of said each distributed processing unit is a percentage of saturation of said each distributed processing unit.

21. (Original) The data processing system as claimed in claim 19, wherein said each distributed processing unit is programmed for collecting utilization statistics of said each distributed processing unit.

22. (Original) The data processing system as claimed in claim 19, wherein the processor is programmed for collecting utilization statistics from said each distributed processing unit.

23. (Original) The data processing system as claimed in claim 19, wherein the respective weight for said each distributed processing unit is programmed into a mapping table, and the processor is programmed to apply the mapping function to the respective utilization value of said each distributed processing unit to obtain a respective weight for said each distributed processing unit by indexing the mapping table with said each respective utilization value of said each distributed processing unit to obtain the respective weight for said each distributed processing unit.

24. (Original) The data processing system as claimed in claim 19, wherein the mapping function is programmed to produce weights estimated to cause a balancing of loading upon the distributed processing units.

25. (Original) The data processing system as claimed in claim 19, wherein the processor is programmed for using the respective weights for weighted round-robin load balancing of the work requests upon the distributed processing units.

26. (Original) The data processing system as claimed in claim 19, wherein the processor is programmed for performing round-robin load balancing of the work requests upon the distributed processing units when the weights are equal.

27. (Original) The data processing system as claimed in claim 19, wherein the processor is programmed for using the respective weights for the distributed processing units for distributing work requests to the distributed processing units by creating a distribution list containing entries indicating the distributed processing units, the respective weight for said each distributed processing unit specifying the number of the entries indicating said each distributed processing unit, and by randomizing the distribution list, and accessing the randomized distribution list for distributing the work requests to the distributed processing units as indicated by the entries in the randomized distribution list.

28. (Currently amended) The data processing system as claimed in claim [[19]] 37, wherein the processor is programmed for re-randomizing the distribution list for re-use once the end of the distribution list is reached during the distribution of the work requests to the distributed processing units as indicated by the entries in the randomized distribution list.

29. (Original) A data processing system comprising distributed processing units and a processor coupled to the distributed processing units for distributing work requests to the distributed processing units, the processor being programmed for:

obtaining a respective utilization value of each distributed processing unit;

applying a mapping function to the respective utilization value of said each distributed processing unit to obtain a respective weight for said each distributed processing unit;

using the respective weights for the distributed processing units for producing a distribution list for distributing work requests to the distributed processing units for load balancing of the work requests upon the processing units, and

repetitively randomizing the distribution list during the distribution of the work requests to the distributed processing units.

30. (Original) A data processing system comprising virus checking servers and a network file server coupled to the virus checking servers for distributing virus checking requests to the virus checking servers, the network file server being programmed for:

obtaining a respective utilization value of each virus checking server, the respective utilization value of said each virus checking server indicating a percentage of saturation of said each virus checking server;

applying a mapping function to the respective utilization value of said each virus checking server to obtain a respective weight for said each virus checking server; and

using the respective weights for the virus checking servers for weighted round-robin load balancing of virus checking requests from the network file server to the virus checking servers.

31. (Original) The data processing system as claimed in claim 30, wherein said each virus checking server is programmed for collecting statistics for calculating the respective utilization value of said each virus checking server.

32. (Original) The data processing system as claimed in claim 30, wherein the respective weight for said each virus checking server is programmed into a mapping table, and the network file server is programmed for indexing the mapping table with said each respective utilization value of said each virus checking server to obtain the respective weight for said each virus checking server.

33. (Original) The data processing system as claimed in claim 30, wherein the network file server is programmed for performing round-robin load balancing of the virus checking requests upon the virus checking servers when the weights are equal.

34. (Original) The data processing system as claimed in claim 30, wherein the network file server is programmed for using the respective weights for the virus checking servers for weighted round-robin load balancing of virus checking requests from the network file server to the virus checking servers by creating a distribution list containing entries indicating the virus checking servers, the respective weight for said each virus checking server specifying the number of the entries indicating said each virus checking server, and by randomizing the distribution list, and accessing the randomized distribution list for distributing the virus checking

requests from the network file server to the virus checking servers as indicated by the entries in the randomized distribution list.

35. (Original) The data processing system as claimed in claim 34, wherein the network file server is programmed for re-randomizing the distribution list for re-use once the end of the distribution list is reached during the distributing of the work requests to the virus checking servers as indicated by the entries in the randomized distribution list.

36. (Original) The data processing system as claimed in claim 34, wherein the network file server is programmed for collecting utilization statistics from the virus checking servers at the start of a heartbeat interval, for randomizing the distribution list repetitively during use of the distribution list for load balancing of virus checking requests during the heartbeat interval, for collecting a new set of utilization statistics from the virus checking servers at the start of a following heartbeat interval, and for producing a new distribution list from the new set of utilization statistics for load balancing of virus checking requests during the following heartbeat interval.